**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Сибирский государственный университет науки и технологий**

**имени академика М.Ф. Решетнева»**

Институт инженерной экономики, прикладная информатика в экономике

институт/ факультет/ подразделение

Кафедра информационных экономических систем

кафедра/ цикловая комиссия

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

Разработка проекта диаграмм классов и диаграмм динамики (Sequense Diagram) для разрабатываемого приложения

тема

Вариант №15

Преподаватель И. В. Василенко

подпись, дата ФИО

Обучающийся БПЭ23-01, 23140116 И. С. Примеров

подпись, дата ФИО

Красноярск 2025

**Цель работы:** узнать внутренние структуры диаграммы классов и диаграммы последовательностей, освоить основные принципы их работы и создать диаграммы, соответствующие варианту задания в Modelio Open Source.

**Ход работы.**

Создадим диаграмму класса в проекте.

Добавим классы и связи на диаграмме. Диаграмма классов «Ресторан» представлена на Рисунке 1.

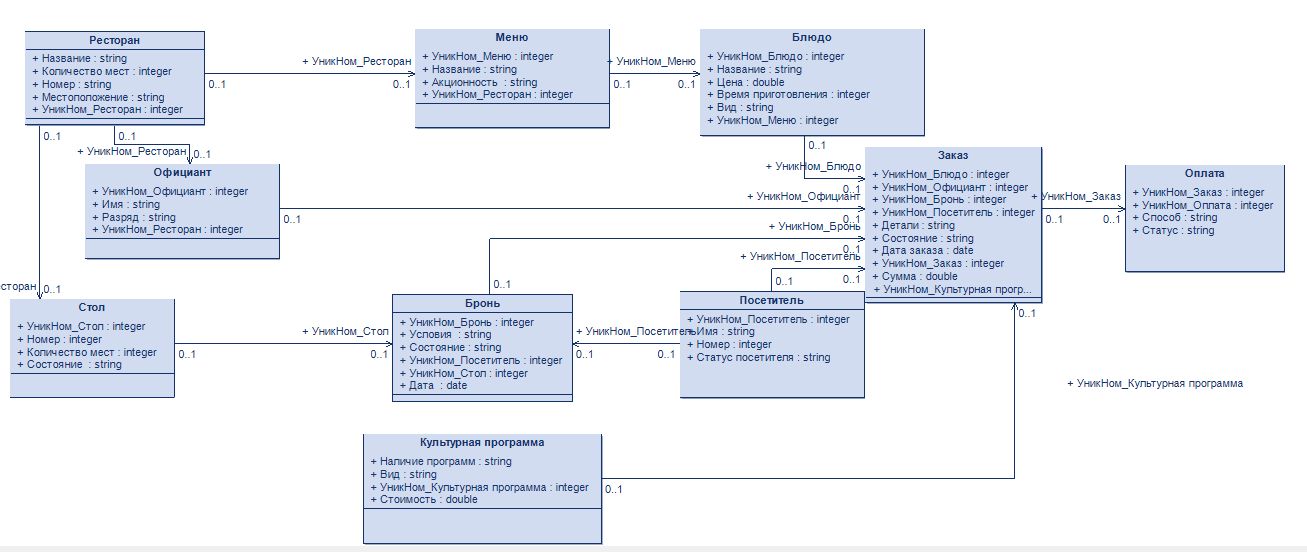


Рисунок 1 – Class Diagram «Ресторан».

**Класс 1: Ресторан**

Описание: содержит основную информацию о ресторане.  
Атрибуты:

* УникНом\_Ресторан: integer — уникальный идентификатор ресторана;
* Название: string — наименование ресторана;
* Номер: string — контактный телефон;
* Количество мест: integer — количество посетителей;
* Местоположение: string — физическое расположение ресторана.

**Класс 2: Меню**

Описание: Описывает меню ресторана.  
Атрибуты:

* УникНом\_Меню: integer — уникальный идентификатор меню;
* Название: string — название меню (например, "Основное", "Ланч-меню");
* Акционность: string — акционные блюда.
* УникНом\_Ресторан: integer — ссылка на ресторан, к которому относится меню.

**Класс 3: Блюдо**

Описание: Содержит информацию о блюдах в меню.  
Атрибуты:

* УникНом\_блюдо: integer — уникальный идентификатор блюда;
* Название: string — наименование блюда;
* Цена: double — стоимость блюда;
* Время\_приготовления: integer — время в минутах;
* УникНом\_Меню: integer — ссылка на меню, в котором представлено блюдо.
* Вид: string ­– вид блюда(Например: горячее, холодное)

**Класс 4: Посетитель**

Описание: Хранит данные о посетителях ресторана.  
Атрибуты:

* УникНом\_Посетитель: integer — уникальный идентификатор клиента;
* Имя: string — имя клиента;
* Номер: string — контактный телефон;
* Статус посетителя: string — Статус посетителя по программе лояльности.

**Класс 5: Официант**

Описание: Содержит информацию об официантах.  
Атрибуты:

* УникНом\_Официант: integer — уникальный идентификатор официанта;
* Имя: string — имя официанта;
* Разряд: integer — разряд официанта исходя из успешно выполненной работы ;
* УникНом\_Ресторан: integer — ссылка на ресторан, где работает официант.

**Класс 6: Бронирование**

Описание: Описывает бронирование стола.  
Атрибуты:

* УникНом\_Бронь: integer — уникальный идентификатор бронирования;
* Дата: date — дата и время бронирования стола;
* Состояние: string — текущий статус бронирования (например, *ожидается*, *подтверждено*, *отменено*);
* УникНом\_Посетитель: integer — ссылка на клиента, сделавшего бронирование;
* УникНом\_Стол: integer — ссылка на зарезервированный стол.
* Условия: string – условия посетителя по бронированию столика.

**Класс 7: Стол**

Описание: Описывает информацию о столах ресторана.  
Атрибуты:

* УникНом\_Стол: integer — уникальный идентификатор стола;
* Номер: integer — номер стола в зале;
* Количество мест: integer — максимальное количество человек за столом;
* Состояние: string — текущий статус стола (например, свободен, занят, забронирован).

**Класс 8: Заказ**

Описание: Фиксирует заказы клиентов.  
Атрибуты:

* УникНом\_Заказ: integer — уникальный идентификатор заказа;
* Дата\_заказа: date — дата и время оформления;
* Состояние: string — текущий статус (принят, готовится, выполнен);
* Сумма: double — общая стоимость заказа;
* УникНом\_Посетитель: integer — ссылка на посетителя;
* УникНом\_Официант: integer — ссылка на официанта;
* УникНом\_Бронь: integer — ссылка на бронирование;
* УникНом\_Блюдо: integer — ссылка на основное блюдо в заказе.
* Детали: string – Основные моменты заказа.
* Культурная программа: integer - ссылка на культурную программу.

**Класс 9: Оплата**

Описание: Содержит информацию о платежах.  
Атрибуты:

* УникНом\_Оплата: integer — уникальный идентификатор оплаты;
* Способ: string — метод оплаты (карта, наличные);
* Статус: string — статус оплаты (успешно, отклонена);
* УникНом\_Заказ: integer — ссылка на соответствующий заказ.

**Класс 10: Культурная программа**

Описание: Содержит информацию о культурных программах.  
Атрибуты:

* УникНом\_Культурная программа: integer — уникальный идентификатор культурной программы;
* Стоимость: double — окончательная цена на программу
* Вид: string —вид программы(Исполнение песни, Танцы);
* Наличие программ: string – Список программ, которые возможно заказать в определенный день.

Далее создадимдиаграмма динамики (Sequence Diagram) «Ресторан» (Рисунок 2).

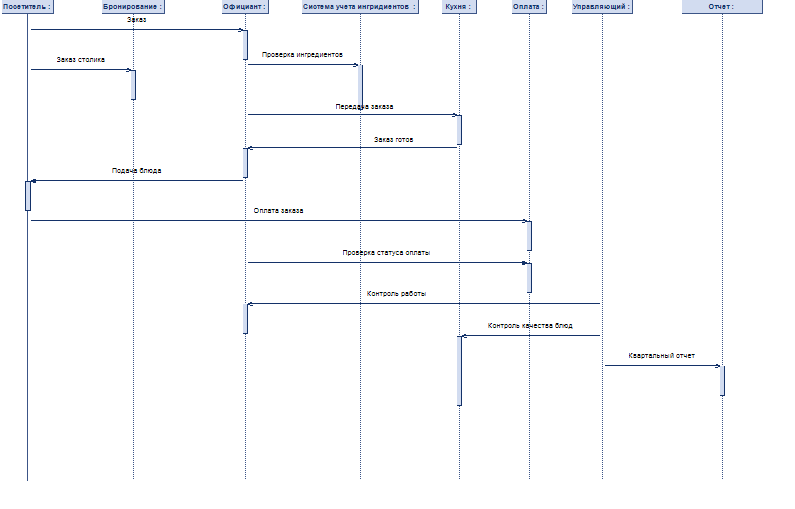


Рисунок 2 – Interaction *Sequence Diagram* «Ресторан»

Связи:

1. **Посетитель → Бронирование:**
   * Стрелка с текстом *"Заказ столика"*.
2. **Посетитель → Официант:**
   * Стрелка с текстом *"Заказ"*.
3. **Официант → Система (меню):**
   * Стрелка с текстом *"Проверка ингредиентов"*.
4. **Официант → Кухня:**
   * Стрелка с текстом *"Передача заказа"*.
5. **Кухня → Официант:**
   * Стрелка с текстом *"Заказ готов"*.
6. **Официант → Посетитель:**
   * Стрелка с текстом *"Подача блюда"*.
7. **Посетитель → Оплата:**
   * Стрелка с текстом *"Оплата заказа"*.
8. **Оплата → Администратор:**
   * Стрелка с текстом *"Подтвердить оплату"*.
9. **Управляющий → Официант:**
   * Стрелка с текстом *"Контроль работы"*.
10. **Управляющий → Кухня:**
    * Стрелка с текстом *"Контроль качества блюд"*.
11. **Управляющий → Отчет:**
    * Стрелка с текстом *"Квартальный отчет"*.

**Вывод**

В ходе выполнения лабораторной работы была смоделирована работа бизнес-единицы «Ресторан» с использованием диаграммы классов и диаграммы динамики. Были отражены ключевые участники, взаимодействия и этапы процесса обработки Ресторана. Построенные модели позволяют наглядно представить структуру и логику процесса, что способствует его систематизации и повышению эффективности.